Simulated Annealing algorithm for VLSI floorplanning for soft blocks

Rajendra Bahadur Singh
Dept. of Electronics & Comm., School of ICT
Gautam Buddha University,
Greater Noida, INDIA
rbs2006vlsi@gmail.com

Anurag Singh Baghel
Dept. of Computer Science, School of ICT
Gautam Buddha University,
Greater Noida, INDIA
asb@gbu.ac.in

Abstract— **In the VLSI physical design, Floorplanning is the very crucial step as it optimizes the chip. The goal of floorplanning is to find a floorplan such that no module overlaps with other, optimize the interconnection between the modules, optimize the area of the floorplan and minimize the dead space. In this Paper, Simulated Annealing (SA) algorithm has been employed to shrink dead space to optimize area and interconnect of VLSI floorplanning problem. Sequence pair representation is employed to perturb the solution. The outcomes received after the application of SA on different benchmark files are compared with the outcomes of different algorithms on same benchmark files and the comparison suggests that the SA gives the better result. SA is effective and promising in VLSI floorplan design. Matlab simulation results show that our approach can give better results and satisfy the fixed-outline and non-overlapping constraints while optimizing circuit performance.**

Keywords— ***VLSI Floorplanning, Simulated Annealing Algorithm, Sequence Pair, Dead Space, etc.***

I. INTRODUCTION

Floorplanning[1-3] is an essential design step of VLSI physical design automation[4]. It determines the size, shape, and locations of modules in a chip and as such, it estimates the total chip area, interconnects, and, delays. Computationally, it is an NP-hard problem; researchers have suggested various heuristics and metaheuristic algorithms solve it. A primary research problem in the VLSI floorplanning[12] is its representation. The representation of Floorplan determines the size of search space and the complexity of the transformation between a representation and its corresponding floorplan.

Floorplanning is defined as the process of placing circuit blocks to a given 2D boundary. It is a very crucial step in the VLSI physical design, and the quality of floorplanning significantly affects the successive design steps such as placement and routing. In the early days, floorplanning was tractable since the sizes of chips were limited and designers were able to generate desirable floorplans manually. But in recent years the complexity of design become larger and the number of modules also increased so modern floorplan design becomes impossible manually. Computationally, it is an NP-hard problem [21]; researchers have suggested various heuristics and metaheuristic algorithms solve it. A primary research problem in the VLSI floorplanning is its representation. The representation of floorplan determines the size of search space and the complexity of the transformation between a representation and its corresponding floorplan.

From the complexity point of view, it is an NP-hard problem. The search space increases exponentially with the increase in the number of modules therefore to get an optimum solution is an outdaring task. The quality of floorplanning depends on how it is represented. The representations of floorplans determine the size of the search space and the complexity of transformation between its representation and its corresponding floorplan. There are various representation methods for floorplan such as Bounded Sliced Grid (BSG)[11], Corner Block

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

List (CBL), Transitive Closure Graph (TCG), B-tree[4], O-tree, Sequence Pair, etc. In this paper, simulation has been done for soft blocks.

The constraints of this issue are two-fold:
(i) All blocks should be kept into a given 2D boundary.
(ii) There is no overlap between any two blocks.

The objective is to optimize physical quantities such as area, wirelength, time, noise, voltage, and temperature. These physical quantities have sagacious effects on the properties of a chip. In this paper, we performed the simultaneous minimization of area, wirelength, and timing.

In SA, we use two types of data structures to solve the optimization of floorplanning: interior and constraint graph. The purpose of the interior structure is to store the lower left coordinate of each module so that modules can be placed without any overlapping.

Constraint graph is a representation method, used for floorplan. Here Sequence pair constraint graph is used. This graph represents orders of modules placement

II. PROBLEM STATEMENT

VLSI floorplan is to arrange the modules on a chip, so the inputs for the floorplanning is a set of m rectangular modules S= {b1,b2,b3……bm} with corresponding areas {a1,a2,a3….am}. Widths, heights, and areas of the modules are denoted as wi, hi, and ai ¬respectively, 1≤ i≤ m. The objective of floorplanning is to arrange the modules in such a way, no two modules overlap each other and the area, wirelength, and other performance indices are optimal.

All modules must be in the rectangular frame and/or square frame. In Figure 1, all modules must be packed without any violation of constraints as mentioned above.

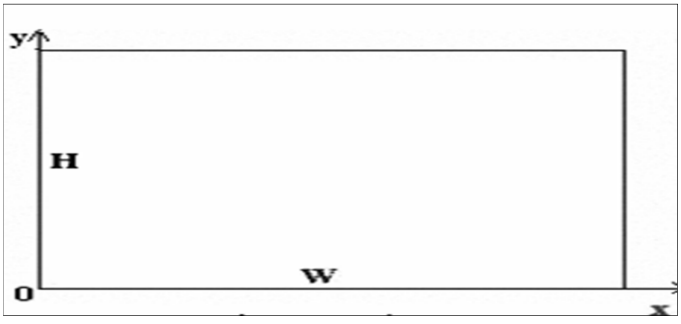


Figure 1. Representation of floorplan

Modules are basically two types, namely hard modules and soft modules. A module is called hard module if its area and an aspect ratio are fixed. A module is called Soft modules if its area is fixed but aspect ratio may vary. The ratio of width and the height of the module are known as the aspect ratio. The floorplan layout can be classified into two types:

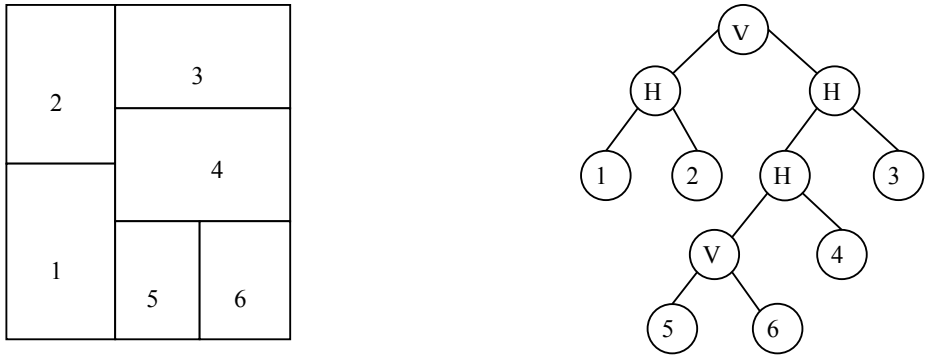
https://sites.google.com/site/ijcsis/
ISSN 1947-5500

A. Sliceable Floorplan

These floorplans can be isolated recursively until each part comprises of a solitary module. A sliceable floorplan is a floor plan that might be characterized recursively as portrayed underneath. Its representation is shown in figure 2..

A floorplan that comprises of a solitary rectangular piece is sliceable.

If a piece from a floorplan can be chopped down in two by a vertical or horizontal line, the subsequent floorplan is sliceable. They can be spoken to by a parallel tree, called the Slicing Tree, having modules at the leaves and cut sorts shown by H (for horizontal cut) and V (for vertical cut) at the internal nodes.



(a) *(b)*

Figure 2. Slicing Floorplan and its tree representation

B. Non-Slicing Floorplan

The non-slicing floorplan can't be cut repetitively. Figure 3, shows a non-slicing floorplan[5] [19].

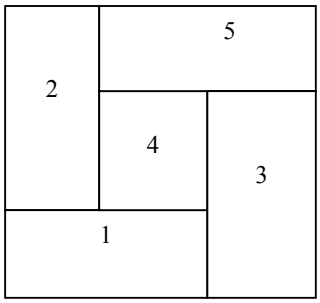


Figure 3. Non-Slicing Floorplan

Floorplaning Cost:

The objective of floorplanning is to arrange the modules in such a way, final output of floorplan structure take the minimal area and minimal interconnect. So the floorplanning cost expression in terms of objective minimal area and minimal interconnect is expresses as

$$Cost(F) = \alpha\left(\frac{Area(F)}{Area*}\right) + (1-\alpha)\left(\frac{Wirelength(F)}{Wirelength*}\right) \quad (1)$$

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Where Area (F) is an area of smallest rectangle enclosing all modules. Wirelenth(F) is interconnection cost between modules. Area*, and Wirelength* is estimated minimum area, and minimum wirelength respectively. Where α is a weight, $0 \leq \alpha \leq 1$.

The objective of the floorplanning problem is to minimize the cost (F). Minimizing of area is achieved if total dead space is minimized.

Dead Space: It is unused area which is not occupied by any module. The dead space calculation formula is:

$$Dead\ Space(D) = \frac{Optimal\ FP\ Area - sum\ of\ all\ modules\ area}{Optimal\ FP\ Area} * 100 \quad (2)$$

Optimal FP area is the product of maximum height and maximum width of the floorplan.

Wirelength Estimation: Half-perimeter wirelength is calculated for all the nets to estimate total wirelength required on the chip. Wirelength between two modules can be computed as follows:

$$L = Xmax - Xmin + Ymax - Ymin \quad (3)$$

Where, Xmax and Xmin are maximum and minimum value of x-coordinate after placing two modules. Similarly, Ymax and Ymin are the maximum and the minimum of y-coordinate after placing two modules. Thus if floorplan problem has n number of nets then total wirelength on the chip can be computed as

$$Total\ wirelength = \sum_{i=1}^{n} L(i) \quad (4)$$

Problem Description

- Inputs of the problem are a set of modules with fixed geometries and fixed pin positions.
- A set of nets, specifying the interconnections between pins of blocks.
- A set of pins with fixed positions.
- A set of user constraints, e.g., block positions/orientations, critical nets, if any

III. FLOORPLAN REPRESENTATION

A primary research problem in the VLSI floorplanning is its representation. The representation of Floorplan determines the size of search space and the complexity of the transformation between a representation and its corresponding floorplan. Many researchers have suggested many representation[9] schemes for floorplan representation such as Corner Block List, B-Tree, Polish Expression, O-tree, Sequence Pair and etc. In this paper, Sequence pair representation has been used only.

Sequence Pair

The first time concept of Sequence pair was given by Murata[13]. It is the most flexible representation of all the representations. A sequence-pair (r+, r-) for an arrangement of m modules is a couple of groupings of the m module names. A sequence-pair imposes horizontal/vertical (HV) constraints for every pair of modules. A sequence pair makes it easier to represent the candidate solution of the stochastic algorithm such as genetic algorithm [6-8]. and simulated annealing. To construct a Sequence Pair, first of all, the Constraint Graph Pair i.e. the HCG and VCG have to be created. For instance, (P=[6 2 5 4 1 3], N=[5 6 1 4 2 3]) represent sequence pair of a floorplan of the six modules namely: 1, 2, 3, 4, 5, and 6.

Where P and N is the random sequence of number of modules.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Sequence Pair is a succinct representation of non-slicing floorplan[19] of the modules.

Lemma 1: For a given sequence pair if module a and b is present in (P, N):

1) Module a is right to module b in floorplan, if a is after b in both P and N sequence.
2) Module a is left to module b in floorplan, if a is before b in both P and N sequence.
3) Module a is above to module b in floorplan, if a is before b in P and is after b in N sequence.
4) Module a is below to module b in floorplan, if a is after b in P and is before b in N sequence.

where P and N is the random sequence of the number of modules.

Example 1: P = [6 2 5 4 1 3], N = [5 6 1 4 2 3]

Table I

Module No.	Left of	Right of	Below	Above
1	[3]	[5, 6]	[2, 4]	Nil
2	[3]	[6]	Nil	[1, 4, 5]
3	Nil	[1, 2, 4, 5, 6]	Nil	Nil
4	[3]	[5, 6]	[2]	[1]
5	[1, 3, 4]	Nil	[2, 6]	Nil
6	[1, 2, 3, 4]	Nil	Nil	[5]

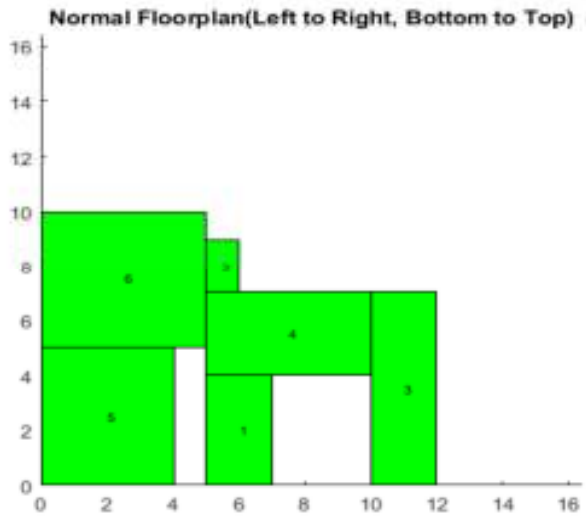


Figure 4. Floorplan representation for example 1 using Sequence pair

Perturb Solution

There are three possible moves in sequence pairs (P, N) to perturb solution to find the better optimal solution.

1) Swapping a random pair of modules in the sequence either P or N.
2) Swapping the random pair of the modules in both sequences i.e. P and N.
3) Rotating a randomly selected module by 90 degrees.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

IV. FLOORPLAN ALGORITHMS

As discussed earlier in this paper, floorplanning determines the positions of modules so that objectives like optimization [17] of area and total interconnect can be achieved. We will discuss Simulated Annealing Algorithm here.

Simulated Annealing (SA)

SA [14] is inspired by an analogy between the physical annealing of solids (crystals) and combinatorial enhancement. In the physical annealing process a solid is first melted and then cooled very slowly, spending a long time at low temperatures, to obtain a perfect lattice structure corresponding to a minimum energy state. SA transfers this process to local search algorithms for combinatorial enhancement problems. It does so by associating the set of solutions of the problem attacked with the states of the physical system, the objective function with the physical energy of the solid, and the optimal solutions with the minimum energy states.

Simulated Annealing [4] is widely known algorithm, it can be effectively apply in the VLSI physical design [17] shown in figure 5 and other fields. Before applying it in VLSI floorplanning for optimum results, need to concern four ingredients.

1) Solution Space
2) Neighborhood structure
3) Cost function
4) Annealing Structure

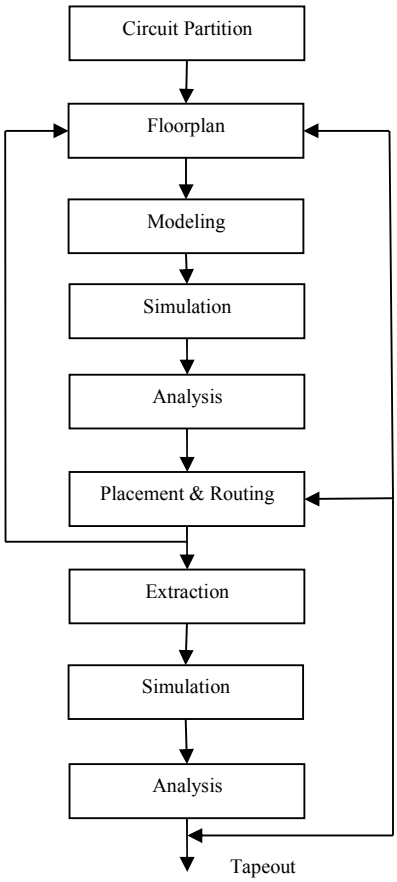


Figure 5. VLSI Physical Design

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

General algorithmic outline for SA:

Procedure Simulated Annealing Algorithm

1. Input benchmarks circuits.
2. Generate randomly an initial solution.
3. Initialize Annealing parameters such as Temperature, Number of iterations, cooling rate etc.
4. Perturb the initial solution and generate a new solution.
5. If new solution result is better than old solution then replace the old solution with new solution.

Otherwise, accept the new solution with probability $e^{-\Delta C/T}$

6. If the number of iterations at the current temperature reaches the length of Markov chain then switch to next step, otherwise move to step 4.
7. Decrease the temperature by cooling rate(α) i.e,

T(i)=α.T(i-1)

8. Repeat the step from 4 to 7 until reach to termination criteria.

Where T= initial temperature, Cost ($\Delta C) = new\ cost - old\ cost$

SA performs computation that analogous to physical process:

- The energy corresponds to the cost function.
- Molecular movement corresponds to a sequence of moves in the set of feasible solution.
- Temperature corresponds to a control parameter T, which control the acceptance probability for a move i.e. a good move.

V. PERFORMANCE ANALYSIS AND SIMULATION RESULTS

The proposed Simulated Annealing algorithm is implemented in MATLAB programming language, and the experiment was executed on an Intel(R) Core(TM) 2 Duo CPU(2.4GHz, 4GB RAM) machine running windows 2007. The parameters of SA algorithm were set as follows, initial temperature=2000, cooling rate (α) =0.95, number of iterations=20.

The proposed Simulated Annealing algorithm is tested with one of the benchmark circuits named as MCNC (Microelectronic Centre of North Carolina) and find the solutions of modern floorplanning problems [18]. These MCNC benchmarks are standard problems in VLSI floorplanning. The details of MCNC benchmark circuits are shown in table I.

TABLE I. DETAILS OF MCNC BENCHMARK CIRCUITS

S.No.	Problem	Benchmarks	No.of modules	No. of nets	No.of pins
1	Apte	MCNC	9	97	287
2	Xerox	MCNC	10	203	698
3	HP	MCNC	11	83	309
4	Ami33	MCNC	33	123	522

TABLE II. SIMULATON RESULT FOR MCNC BENCHMARKS

Benchmark	Area	HPWL	Dead Space %	Simulation Time(Sec)
Apte	47.35	663754	1.68	35
Xerox	19.66	575710	1.61	49
HP	9.11	227224	3.16	29
Ami33	1.20	101753	3.62	107

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

TABLE III. COMPARISION OF AREA WITH DIFFERENT ALGORITHM

Benchmark	VOAS[21]	PSO[20]	TCG[10]	SA (our)
Apte	47.1	47.31	46.92	47.35
Xerox	20.3	20.38	19.83	19.66
HP	9.46	--	8.94	9.11
Ami33	1.20	1.29	1.20	1.20

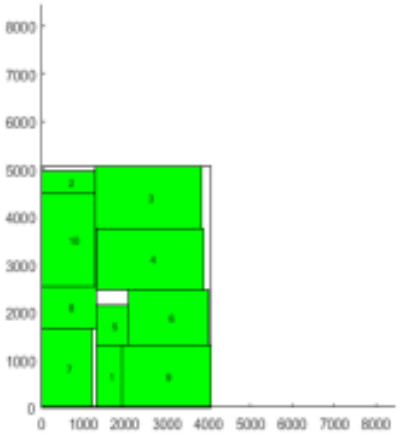


Figure 6. Floorplan Layout for Xerox benchmark

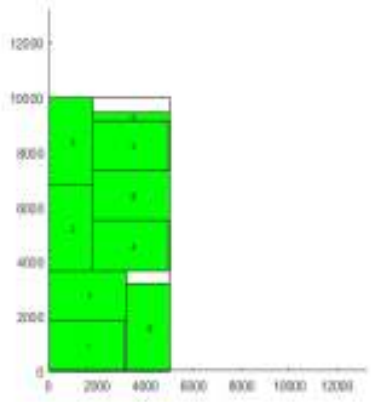


Figure 7. Floorplan Layout for Apte benchmark

Xerox benchmark file contains 10 modules. Using the Simulated Annealing, it takes 20 iterations to improve the performance. Here the total area minimized to 19.66, and the dead space minimized to 1.61%. The Simulated result for "Xerox"file is shown in figure 6.

Apte benchmark file contains 9 modules. Using the Simulated Annealing, it takes 20 iterations to improve the performance. Here the total area minimized to 47.35, and the dead space minimized to 1.6 %. The Simulated result for "Apte" file is shown in figure 7.

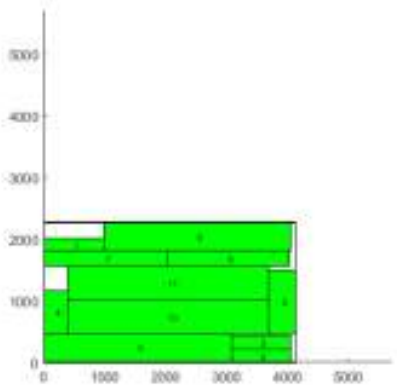


Figure 8. Floorplan Layout for HP benchmark

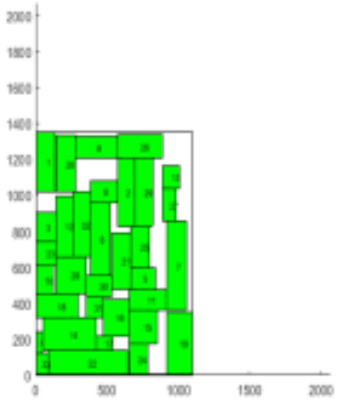


Figure 9. Floorplan Layout for AMI33 benchmark

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

HP benchmark file contains 11 modules. Using the Simulated Annealing, it takes 20 iterations to improve the performance. Here the total area minimized to 9.11, and the dead space minimized to 3.16. The Simulated result for "HP"file is shown in Figure 8.

Ami33 benchmark file contains 33 modules. After 20 iterations, Simulated Annealing Algorithm has improved the performance. Here the total area minimized to 1.20, and the dead space minimized to 3.62. The Simulated result for "Ami33"file is shown in Figure 9.

VI. CONCLUSION

A Simulated Annealing (SA) algorithm is proposed to tackle VLSI floorplanning problem. In this paper we conclude that optimization of VLSI Floorplanning using simulated annealing with Sequence pair representation gives better result for MCNC benchmarks. From table III, it is obvious that using SA algorithm area optimization is less compare to other algorithms. SA reduces the area but it takes more iteration and more computational time to find optimal solution. Further research on the application of SA in VLSI floorplanning design problem is to minimization of power and thermal optimization can be do. For future work, we have to study more algorithms like Particle Swarm Optimization algorithm, Genetic algorithm [15][16], Ant Colony Optimization[21], Cuckoo search algorithm to solve modern VLSI floorplanning problem with less computational time along with reshaping layout of circuit.

REFERENCES

[1] Kurbel, Karl, Bernd Schneider, and Kirti Singh. "Solving optimization problems by parallel recombinative simulated annealing on a parallel computer-an application to standard cell placement in VLSI design." Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 28.3 (1998): 454-461.
[2] Nakaya, Shingo, Tetsushi Koide, and Si Wakabayashi. "An adaptive genetic algorithm for VLSI floorplanning based on sequence-pair." Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on. Vol. 3. IEEE, 2000.
[3] C. J. Alpert and D. P. Mehta, Handbook of algorithm for physical design automation", Auerbach Publications, pp. 139-142, 2008.
[4] Lichen, Zhu, et al. "An efficient simulated annealing based VLSI floorplanning algorithm for slicing structure." Computer Science & Service System (CSSS), 2012 International Conference on. IEEE, 2012.
[5] Leena Jain and Amarbir Singh, "Non Slicing Floorplan Representations in VLSI Floorplanning: A Summary," International Journal of Computer Applications (0975 – 8887), vol. 71, No. 15, June 2013.
[6] J. P. Cohoon and W. D. Paris: "Genetic placement," Proc. IEEE Int. Conf. on CAD, pp.422425 (1986).
[7] L. Davis(ed.): "Handbook of Genetic Algorithms," Van Nos- trand Reinhold (1991).
[8] D. E. Goldberg: "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley Publishing Company (1989)
[9] Laskar, Naushad Manzoor, et al. "A survey on VLSI Floorplanning: Its representation and modern approaches of optimization." Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on. IEEE, 2015.
[10] Jai -Ming Lin and Yao-Wen Chang, "TCG: A Transitive Closure Graph-based Representation for Non Slicing Floorplan", Proceedings of Design Automation Conference , pp. 764-769, 2001.
[11] S. Nakatake, K. Fujiyoshi, H. Murata and Y. Kajitani, "Module placement on BSG-structure and IC layout applications," Proceedings of 1996 IEEE/ACM, International Conf. on Computer Aided Design, 1996, pp. 484-491.
[12] Singh, Rajendra Bahadur, Anurag Singh Baghel, and Ayush Agarwal. "A review on VLSI floorplanning optimization using metaheuristic algorithms." Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on. IEEE, 2016.
[13] H. Murata, K. Fujiyoshi, and Y.Kajitani, "VLSI module placement based on rectangle-packing by the sequence-pair," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 15, No. 12, 1996, pp. 1518-1524.
[14] Chen, Tung-Chieh, and Yao-Wen Chang. "Modern floorplanning based on B/sup*/-tree and fast simulated annealing." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 25.4 (2006): 637-650.
[15] B. H. Gwee, and M. H. Lim, "A GA with heuristic-based decoder for IC Floorplanning," Integration, the VLSI journal 28, 1999, pp. 157-172.
[16] X. G. Wang, L. S. Yao, and J. R. Gan, "VLSI Floorplanning Method Based on Genetic Algorithms," Chinese Journal of Semiconductors, 2002, pp. 330-335.
[17] D.Jackuline Moni, "Certain Optimization techniques for Floorplanning in VLSI Physical Design," PhD Thesis, Faculty of Information and Communication Engineering, Anna University, June 2009.
[18] Jai-Ming Lin, and Zhi-Xiong Hung, SKB-Tree: A Fixed-Outline Driven Representation for Modern Floor-planning Problems" IEEE transactions on VLSI systems, vol: 20, issue: 3, pp: 473-484, 04 Feb 2011.
[19] Y. C. Chang, Y. W. Chang, G. M. Wu, and S. W.Wu,"B*-Trees: A new representation for nonslicing floorplans," Design Automation Conference, pp. 458-463. 2000.
[20] Guolong Chen, Wenzh ong Guo and Yuzhong Chen, " A PSO -based Intelligent Decision Algorithm for VLSI Floorplanning", Journal of Soft Computing A Fusion of Foundation, Methodologies and Applications, Vol. 14, No. 12, pp. 1329 -1337, 2010.
[21] Chyi Shiang Hoo, Kanesan Jeevan , Velappa Ganapathy and Harikrishnan Ramiah , "Variable Order Ant System for VLSI Multi Objective Floorplanning", Applied Soft Computing, Vol. 13, No. 7, pp. 3285-3297, 2013.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500